Data Aggregation and Demand Prediction

Maxime C. Cohen
McGill University, Montreal, Canada, maxime.cohen@mcgill.ca

Kevin Jiao
NYU Stern School of Business, New York, NY 10012, jjiao@stern.nyu.edu

Renyu Zhang
NYU Shanghai, 1555 Century Avenue, Shanghai, China, 200122, renyu.zhang@nyu.edu

Retailers collect large volumes of transaction data with the goal of predicting future demand. We study how retailers could use clustering techniques to improve demand prediction accuracy. High accuracy in demand prediction allows retailers to better manage their inventory, and ultimately mitigate stock-outs and excess supply. It is thus important for retailers to leverage their data for demand prediction. A typical retail setting involves predicting demand for hundreds of products simultaneously. While some products have a large amount of historical data, others were recently introduced and transaction data can be scarce. A common approach is to cluster several products together and estimate a joint model at the cluster level. In this vein, one can estimate some model parameters by aggregating the data from several items, and other parameters at the item level. In this paper, we propose a practical method—referred to as the Data Aggregation with Clustering (DAC) algorithm—that balances the tradeoff between data aggregation and model flexibility. The DAC allows us to predict demand while optimally identifying the features that should be estimated at the (i) item, (ii) cluster, and (iii) aggregated levels. We analytically show that the DAC yields a consistent estimate along with improved asymptotic properties relative to the traditional ordinary least squares method that treats different items in a decentralized fashion. Using both simulated and real data, we illustrate the improvement in prediction accuracy obtained by the DAC relative to several common benchmarks. Interestingly, the DAC not only has theoretical and practical advantages, it also helps retailers discover useful managerial insights.

Key words: Retail analytics, demand prediction, data aggregation, clustering

1. Introduction

Retailers routinely collect large volumes of historical data. These data are used to improve future business practices such as inventory management, pricing decisions, and customer segmentation. One of the most important data-driven task for a retailer is to predict the demand for each stock keeping unit (SKU). A common approach in practice is to split the SKUs in departments (e.g., soft drinks), and sometimes even in sub-categories (e.g, a specific format of soft drinks).

Predictive models for demand prediction have been extensively studied and applied in practice. A typical demand model is a regression specification with the sales (or logarithmic of the sales) as the outcome variable, and price, seasonality, brand, color, and promotion as independent variables or features. The model coefficients are then estimated using historical data.

In many retail settings, a subset of items have been offered for a long time, whereas other items were recently introduced. For such newly introduced items, only a limited number of historical observations is available. It is thus crucial to reduce the dimensionality of the feature space in order to decrease the variance of the estimated coefficients. At the same time, the SKUs in the same department often share similar characteristics, and hence tend to be affected by a particular feature in a similar way. A prominent approach is to estimate certain coefficients at the aggregate level (i.e., by gathering the data across all SKUs and assuming a uniform coefficient). For example, it seems reasonable to believe that all the items in the ice-cream category share the same seasonality pattern. Although this approach has been widely adopted in the retail industry, no rigorous empirical method has been developed to formalize how this data aggregation procedure should be applied for demand prediction. In this paper, we seek to bridge this gap by formalizing the tradeoff between data aggregation (i.e., finding the right level of aggregation for each coefficient) and model flexibility (i.e., estimating a different model for each item) in a systematic fashion.

Due to insufficient data, the traditional approach of estimating a different model for each SKU is usually inefficient for new products or for SKUs with noisy observations. This approach cannot identify the right aggregation level for each coefficient, and does not find the underlying cluster structure of the coefficients. Based on common clustering methods (e.g., k-means), we propose an efficient and integrated approach to infer the coefficient of each feature while identifying the right level of data aggregation based on the statistical properties of the estimated coefficients. Our method also allows us to incorporate multiple aggregation levels while preserving model interpretability. From a practical perspective, our method can be easily estimated using retail data and yields a significant improvement in out-of-sample prediction accuracy.

1.1. Main Results and Contributions

We study the tradeoff between data aggregation and model flexibility by optimally identifying the right level of aggregation for each feature as well as the cluster structure of the items. We propose a practical method—referred to as the Data Aggregation with Clustering (DAC) algorithm—that allows us to predict demand while optimally identifying the features that should be estimated at the (i) item, (ii) cluster, and (iii) aggregated levels. Our proposed algorithm first applies the maximum likelihood estimation approach to estimate a different coefficient vector for each item (called the decentralized model). It then performs a hypothesis test (i.e., $t-$test) on the estimated coefficients from the decentralized model to identify the right aggregation level of each feature. To characterize the cluster structure of the items, we apply the $k-$means method on the estimated coefficients from the decentralized model (as opposed to the features themselves).

We first characterize the theoretical properties of the DAC. Specifically, we show that the DAC yields a consistent estimate of the (i) data aggregation levels, (ii) cluster structure, and (iii) feature coefficients. As a result, if the data has enough observations, one can correctly identify the “true” data generating process. In addition to this consistency result, we demonstrate improved asymptotic properties—smaller variance and a tighter probabilistic bound—relative to the commonly used ordinary least squares method. Armed with these theoretical results, we next conduct several computational experiments—based on both simulated and real data—to illustrate the significant improvement of the DAC in (out-of-sample) prediction accuracy relative to several common benchmarks. Our results highlight the essential value of the proposed DAC algorithm in better balancing the bias-variance tradeoff, resulting in more accurate demand prediction. Finally, we apply the DAC using two years of retail data and convey that it can also help retailers discover useful insights on the relationships between the different items.

1.2. Related Literature

This paper is related to several streams of literature including (i) prediction and clustering algorithms and (ii) retail operations and demand forecasting.

Prediction and clustering algorithms: The problems of demand prediction and clustering are extensively studied in the machine learning (ML) literature. Bertsimas and Kallus (2014) combine ideas from ML and operations research to develop a new prediction method. The authors solve a conditional stochastic optimization problem by incorporating ML methods such as local regression and random forests. Kao et al. (2009) and Donti et al. (2017) focus on developing new ML methods by training a prediction model with respect to a nominal optimization problem. Although several previous papers study general settings, it is hard to apply existing methods to a retail setting where multiple levels of hierarchy may exist. Elmachtoub and Grigas (2017) propose a new idea called Smart “Predict, then Optimize” (SPO). The key component of SPO is that the loss function is computed based on comparing the objective values generated using predicted and observed data. The authors then address the computational challenge and develop a tractable version of SPO. Jagabathula et al. (2018) propose a model-based embedding technique to improve the clustering algorithm to segment a large population of customers into non-overlapping groups with similar preferences. Liu et al. (2019) apply clustering techniques to predict the travel time of last-mile delivery services and optimize the order assignment for such services.

Our work is also related to the traditional clustering literature. Since the introduction of k-means by MacQueen et al. (1967), clustering algorithms have been extensively studied. In particular, k-means has been widely applied to a variety of domains, such as image segmentation (Marroquin and Girosi 1993). In the context of assortment personalization, Bernstein et al. (2018) have recently proposed a dynamic clustering method to estimate customer preferences. In our paper, we leverage some theoretical properties of the k-means clustering method and embed this clustering method as one of the key steps in our demand prediction algorithm.

Retail operations and demand forecasting: Retailers always seek to improve operational decisions, such as inventory replenishment, supply chain management, and revenue management and pricing. These decisions all closely rely on accurate demand forecasting/prediction. As reported in Cohen and Lee (2019), demand uncertainty is a major issue in designing efficient global supply chains. There is a large body of literature that focuses on developing methods for demand prediction in retail settings. Given the increasing volume of transaction data collected by retailers, sophisticated models have emerged in the past two decades. Marketing papers such as Foekens et al. (1998) and Cooper et al. (1999) estimate econometrics models to draw managerial insights on the impact of retail promotions. In a similar vein, Van Heerde et al. (2000) and Macé and Neslin (2004) study the pre- and post-promotion dips using linear regression models with lagged variables. Kök and Fisher (2007) develop a procedure to estimate substitution behavior in retail demand. Recent developments in demand prediction include the following three papers: Huang et al. (2014) try to embed the competitive information (including price and promotions) into demand prediction, Fildes et al. (2019) suggest that promotional information can be quite valuable in improving forecast accuracy, and Huang et al. (2019) further take into account the impact of marketing activities. In the operations management community, demand prediction models are often used as an input to an optimization problem for supermarkets (see, e.g., Cohen et al. 2017b,a) and for hotels (see, e.g., Pekgün et al. 2013). For example, Cohen et al. (2017b) estimate a log-log demand model using supermarket data. The authors then solve the promotion optimization problem by developing an approximation based on linear programming. It has been shown in the retail operations literature that responding to accurate demand forecasts can substantially increase profits (Caro and Gallien 2010). At nationwide level, Kesavan et al. (2010) show that wisely incorporating cost of goods sold, inventory, and gross margin information can substantially improve firm-level sales forecast for retailers. In recent years, the amount of data available has grown exponentially. It thus offers new opportunities for research in demand prediction (Feng and Shanthikumar 2018). In this context, our paper proposes a new demand prediction method that can efficiently aggregate data from multiple items to improve prediction accuracy.

A recent stream of papers integrate a clustering step into demand prediction. For instance, Baardman et al. (2017) propose an iterative approach to cluster products and leverage existing data to predict the sales of new products. Hu et al. (2017) propose a two-step approach to first estimate the product lifecycle, and then cluster and predict. In our paper, however, the definition of clusters is fundamentally different. Unlike previous work, our clustering is based on the estimated coefficients rather than on the features. Furthermore, our model is flexible enough to account for different levels of data aggregation, whereas in previous studies, all features are essentially estimated at the cluster level. Allowing such flexibility is key to improve demand forecasting.

Structure of the paper. The rest of the paper is organized as follows. In Section 2, we introduce our model and discuss the relevant computational challenges. We then describe the DAC algorithm in Section 3. Our analytical results are presented in Section 4. In Sections 5 and 6, we conduct computational experiments using simulated and real data, respectively. Our conclusions are reported in Section 7. The proofs of our analytical results are relegated to the Appendix.

2. Model

We introduce our demand prediction model under the generalized linear model (GLM) framework. We consider a retail department (e.g., soft drinks, electronics) which comprises n items (or SKUs). Each item has m historical observations (e.g., weekly sales information). We use $Y_{i,j}$ to denote the (log-)sales of item i in observation j ($1 \leq i \leq n$ and $1 \leq j \leq m$). The prediction model of each item includes d features (for simplicity of exposition, we assume that each item has the same number of features and observations). The feature set is denoted by $D := \{1, 2, \cdots, d\}$. We also define $X_{i,j} := (X_{i,j}^1, X_{i,j}^2, \cdots, X_{i,j}^d)' \in \mathbb{R}^d$ as the feature vector for item i and observation j.

An important characteristic of our model is that a feature $l \in D$ may affect the demand/sales of an item at different aggregation levels: (i) SKU, (ii) Cluster, and (iii) Department. More precisely, a feature may have the same impact on all items, captured by a uniform coefficient for all the items in the department. We refer to such features as *shared* (department-level) features, the set of which is denoted by D_s. Alternatively, a feature may have a different impact for different items, captured by different coefficients for different items. We refer to such features as *non-shared* (SKU-level) features, the set of which is denoted by D_n. Finally, we assume that the items follow a cluster structure so that some features have the same impact for items within the same cluster and a different impact for items in a different cluster. This phenomenon is captured by a uniform coefficient for all the items in the same cluster (the coefficients are different across different clusters). We refer to such features as *cluster* (cluster-level) features, the set of which is denoted by D_c. We also assume that the number of clusters k is given but the cluster structure is unknown (one can further apply our proposed algorithm for different values of k). The entire feature set, D, can be written as the union of three disjoint sets of features that affect the demand at different aggregation levels: $D = D_s \cup D_n \cup D_c$. The feature aggregation structure D_s, D_n, and D_c are unknown a priori and should be estimated from data. The underlying cluster structure is also unknown.

In the GLM framework, the observations are generated from an exponential family distribution which includes normal, binomial, and Poisson distributions as special cases. Based on the three aggregation levels of the features, we have:

$$\mathbb{E}[Y_{i,j}] = g^{-1}\left(\sum_{l\in D_s} X_{i,j}^l \beta_l^s + \sum_{l\in D_n} X_{i,j}^l \beta_{i,l}^n + \sum_{l\in D_c} X_{i,j}^l \beta_{\mathcal{C}(i),l}^c\right), \qquad i = 1,\dots,n \text{ and } j = 1,\dots,m. \quad (1)$$

Here, $\mathcal{C}(i) \in \{1,\dots,k\}$ is the cluster that contains item i and $g(\cdot)$ represents the link function that establishes the relationship between the linear predictor and the mean of the outcome variable. Furthermore, we use $\mathcal{C}_u$ to denote items in cluster u, where $u \in \{1,2,\dots,k\}$ and $\{\mathcal{C}_1,\mathcal{C}_2,\dots,\mathcal{C}_k\}$ is a partition of the items $\{1,2,\dots,n\}$. There are many commonly used link functions, and in practice the choice is made depending on the setting. For example, if $Y_{i,j}$ is the number of units sold of item i on observation j, $g(\cdot)$ could be the identity function and, as a result, the model reduces to a linear regression. On the other hand, if $Y_{i,j}$ is a binary variable, $g(\cdot)$ can be a logit function. Likewise, there are other examples of link functions such as logarithmic and inverse squared.

Based on Equation (1), we can characterize the aggregation levels of the three types of features. For a department-level feature l, its coefficient β_l^s is shared among all items. In other words, all items in the department have the same coefficient for this feature. In comparison, for a SKU-level feature l, its coefficient $\beta_{i,l}^n$ varies across the different items (i.e., $\beta_{i,l}^n \neq \beta_{k,l}^n$ for $i \neq k$). Finally, for a cluster-level feature l, all items in the same cluster will have the same coefficient, that is, if $i \in \mathcal{C}(i)$, then the coefficient of $X_{i,j}^l$ is equal to $\beta_{\mathcal{C}(i),l}^c$. Thus, the total number of coefficients in our model is $d_x = n|D_n| + k|D_c| + |D_s|$. Note that the notion of estimating the coefficient of certain features at an aggregated level is common in practice. For example, retailers sometimes estimate the seasonality coefficients at the department level to avoid over-fitting. Also, when estimating the effects of promotions such as cannibalization or halo, one would consider clustering some items together because promotions often have a similar impact on a group of items. For expositional and computational convenience, we make the following assumption throughout the paper.

Assumption 1. *(a) If a feature l is at the SKU level (i.e., $l \in D_n$), then $\beta_{i,l} \neq \beta_{i',l}$ for any two items i and i', that is, a SKU-level feature has a different effect for different items.*

(b) If cluster-level features exist (i.e., $l \in D_c \neq \emptyset$), then $\beta_{i,l} = \beta_{i',l}$ if and only if $\mathcal{C}(i) = \mathcal{C}(i')$, that is, a cluster-level feature has the same effect for items in the same cluster and a different effect for different clusters. Furthermore, each cluster has at least two items.

As we discuss below, our method can easily be adapted to the setting where Assumption 1 is relaxed (Assumption 1 simplifies the exposition by avoiding the situation where two clusters have the same coefficient value). We use $n_{i,l}$ to denote the number of items that share the same coefficient as item i for feature l, that is, $n_{i,l} = 1$ if $i \in D_n$, $n_{i,l} = n$ if $i \in D_s$, and $n_{i,l} = |\mathcal{C}(i)|$ if $i \in D_c$.

Our main goal is to accurately predict the dependent variable Y given the features X, assuming the data generating process in Equation (1). The key challenge lies in correctly estimating three essential aspects of our model: (a) the aggregation level of each feature, (b) the cluster structure, and (c) the coefficient of each feature. Before presenting our proposed estimation method, we first discuss why directly estimating the aggregation levels, cluster structure, and feature coefficients can be challenging. Two intuitive methods come to mind for estimating our model. First, one can use the *constrained maximum likelihood estimation* (constrained MLE). This approach revises the standard MLE by adding the constraint that, for items in the same cluster, the coefficients of the features at the department or cluster levels should be the same. Since we do not know a priori the aggregation level of each feature, the constrained MLE approach will involve solving an optimization problem with non-convex (multiplicative) constraints, which is computationally prohibitive when the number of features is large. We provide a more detailed discussion of the impracticality of this approach for our problem in Appendix A.1.

A second possible approach is via *iterative optimization*, which is essentially one version of the EM (Expectation-Maximization) Algorithm. This approach introduces a binary decision variable to determine the aggregation level of each feature. To simultaneously estimate the aggregation level and the coefficient of the features, we iteratively estimate the aggregation level binary variable and the coefficients using a MLE. The iterative procedure will stop once the binary variables remain unchanged for two consecutive iterations. A similar iterative optimization approach was proposed by Baardman et al. (2017) to address the demand forecasting problem with two feature aggregation levels (SKU and cluster levels). In their setting, this iterative procedure was proved to converge to the true coefficients and aggregation levels (i.e., the estimate is consistent). In our setting, however, the validity of the iterative optimization approach heavily relies on the initialization of the parameters. Depending on the initial parameter, the procedure may reach a local optimum without any guarantee of convergence to a global optimal solution. For more details on the iterative optimization approach in our context, see Appendix A.2. Finally, an important shortcoming of the constrained MLE and the iterative optimization is that neither method can simultaneously identify the cluster structure and estimate the coefficients.

3. Data Aggregation with Clustering

As mentioned, the problem of estimating the feature coefficients, aggregation levels, and cluster structure is computationally challenging and subject to substantial prediction errors. In this section, we propose a novel data aggregation approach which allows us to (a) identify the right level of aggregation for each feature, (b) find the underlying cluster structure, and (c) generate a consistent estimate of the coefficients for the GLM model. Our method is entirely data-driven and could efficiently achieve the aforementioned three goals in an integrated fashion as long as we have sufficiently many observations in the training set (i.e., m is sufficiently large).

We start our analysis by focusing on a (simple) special case of the model in Equation (1). Specifically, we assume that all the features are at the SKU-level. In this case, the data generating process can be written as follows:

$$\mathbb{E}[Y_{i,j}] = g^{-1}\left(\sum_{l\in D} X_{i,j}^l b_{i,l}\right), \qquad i=1,\dots,n \text{ and } j=1,\dots,m. \tag{2}$$

Comparing Equations (1) and (2), we have $b_{i,l} = \beta_l^s$ for $l \in D_s$, $b_{i,l} = \beta_{i,l}^n$ for $l \in D_n$, and $b_{i,l} = \beta_{\mathcal{C}(i),l}^c$ for $l \in D_c$. We refer to model (2) as the *decentralized model.* The estimation of the decentralized model is usually carried out through iterative re-weighted least squares, which ultimately lead to the MLE. We assume that for each item, the decentralized model is well defined and does not lead to multiple MLE solutions. As expected, the estimation of the decentralized model can be decomposed into estimating each item separately. Specifically, using the data of item i, we apply the MLE to obtain the estimated coefficients of item i, $\hat{b}_i := (\hat{b}_{i,1}, \hat{b}_{i,2}, \cdots, \hat{b}_{i,d})' \in \mathbb{R}^d$, as follows:

$$\hat{b}_i \in \arg\max_{b_i} \sum_{j=1}^m \log \mathcal{L}(b_i | Y_{i,j}, X_{i,j}^1, X_{i,j}^2, \dots, X_{i,j}^d),$$

where $\mathcal{L}(b_i | Y_{i,j}, X_{i,j}^1, X_{i,j}^2, \dots, X_{i,j}^d)$ is the likelihood function associated with the data $\{Y_{i,j}, (X_{i,j}^1, X_{i,j}^2, \dots, X_{i,j}^d)\}$ and the coefficient vector $b_i = (b_{i,1}, b_{i,2}, \cdots, b_{i,d})' \in \mathbb{R}^d$. We refer to the estimator $\hat{b} := (\hat{b}_1, \hat{b}_2, \cdots, \hat{b}_n)$ as the *decentralized estimator*. To estimate the aggregation levels, cluster structure, and feature coefficients, we need to find a partition of the vector $\hat{b}_i$ for each item i, to identify the correct level of aggregation for each feature. Before presenting our algorithm in greater detail, we first state the following consistency property of the decentralized estimator $\hat{b}$.

Lemma 1. *The decentralized estimator $\hat{b}$ is consistent, that is, if $m \uparrow +\infty$, we have:*

- $\hat{b}_{i,l} \xrightarrow{p} \beta_l^s$ *for* $l \in D_s$*;*
- $\hat{b}_{i,l} \xrightarrow{p} \beta_{i,l}^n$ *for* $l \in D_n$*;*
- $\hat{b}_{i,l} \xrightarrow{p} \beta_{\mathcal{C}(i),l}^c$ *for* $l \in D_c$*;*

where $\xrightarrow{p}$ *refers to convergence in probability.*

Lemma 1 shows that with sufficiently many observations, we can consistently estimate the feature coefficients using the decentralized MLE. Two issues remain unaddressed with the decentralized estimation: how can we find the right aggregation level for each feature and how can we identify the cluster structure of the items. Furthermore, the decentralized estimator may suffer from overfitting so that it may have a high variance. This follows from the fact that the number of coefficients of the true model—Equation (1)—is strictly less than the number of coefficients generated by the decentralized estimator: $d_x = n|D_n| + k|D_c| + |D_s| < nd$, where $d = |D_n| + |D_c| + |D_s|$.

It is not surprising that the decentralized estimator, $\hat{b}$, is consistent given that the decentralized model has the highest flexibility. As a result, if we have sufficiently many observations for each item, the forecast performance of the decentralized model will be reasonably good. That said, the decentralized model neither captures the aggregation level of each feature nor leverages the cluster structure of the items. As we discuss in Section 5, exploiting the data aggregation can substantially increase the prediction accuracy. Namely, data aggregation helps us reduce the variance of the estimator and addresses the over-fitting issue.

To estimate the aggregation level and the underlying cluster structure based on Equation (1), we next introduce an additional special case of the model in which the data aggregation level and cluster structure are known. We refer to this case as the *aggregated model* and we call its MLE the *aggregated estimator*, which we denote as $\hat{\beta}$. For the aggregated model, we denote $\hat{\beta}^s_l$ as the estimated coefficient for a department-level feature, $\hat{\beta}^n_{i,l}$ for a SKU-level feature, and $\hat{\beta}^c_{\mathcal{C}(i),l}$ for a cluster-level feature. We are now ready to introduce the *Data Aggregation with Clustering* (DAC) algorithm, which allows us to consistently estimate the coefficient of each feature for each item, as well as correctly identify the right aggregation levels and the underlying cluster structure (see Algorithm 1).

Algorithm 1 DAC

```
Input: Estimated coefficient $\hat{b}_{i,l}$ and standard error ($\hat{SE}_{i,l}$) for each item $i$ and feature $l$.
For each feature $l \in D$:
    Fix an item 1. For all other items $i \neq 1$, compute the $p-$value based on the null hypothesis $H^0_{1,i}$
      that $b_{1,l} = b_{i,l}$, that is, the coefficients of feature $l$ are the same for item 1 and item $i$.
    If $H^0_{1,i}$ is not rejected for all items, then feature $l$ should be estimated at the aggregated level.
    If $H^0_{1,i}$ is rejected for some items and validated for others, then feature $l$ should be estimated
      at the cluster level. We then run a one-dimensional $k$-means algorithm on $\{\hat{b}_{i,l} : 1 \leq i \leq n\}$ and
      obtain the resulting clusters $\hat{\mathcal{C}}_1, \hat{\mathcal{C}}_2, \cdots, \hat{\mathcal{C}}_k$.
    If $H^0_{1,i}$ is rejected for all items, then feature $l$ should be estimated at the SKU level.
    Obtain the aggregation level for each feature: $\hat{D}_n$, $\hat{D}_s$, and $\hat{D}_c$.
    Fit an aggregated model to obtain the coefficients $\hat{\beta}$.
Output: (a) Aggregation levels: $(\hat{D}_n, \hat{D}_s, \hat{D}_c)$, (b) cluster structure: $(\hat{\mathcal{C}}_1, \hat{\mathcal{C}}_2, \cdots, \hat{\mathcal{C}}_k)$, (c) Feature coef-
    ficients: $\hat{\beta}$.
```

The DAC is inspired by Lemma 1. By leveraging the consistent estimated parameters obtained from the decentralized model, we can perform a hypothesis testing to identify the right data aggregation levels and cluster structure. Options for hypothesis testing include the Wald test and the likelihood ratio test (see, e.g., Greene 2003). The main idea is that if the estimated coefficients $\hat{b}_{i,l}$ and $\hat{b}_{i',l}$ are statistically close to one another, then it is very likely that either item i and item i' belong to the same cluster or that feature l is an aggregated-level feature. Another interesting characteristics of our method is that it uses the estimated coefficients as inputs to identify the cluster structure of the items (as opposed to item attributes as in traditional clustering algorithms). If some clusters do not agree for different features, one can implement a majority vote to decide the optimal cluster structure. Alternatively, one can pool cluster-level feature coefficients and fit a multi-dimensional k-means. Either approach could produce a consistent estimate of the cluster structure. We note that under Assumption 1, the pairwise hypothesis testing step has a time complexity of $O(n)$.[1] Furthermore, the identification of cluster-level features is even more efficient. Indeed, if we infer that the coefficients of a feature coincide for some items and differ for others, then this feature must be at the cluster-level. Finally, we remark that the last step of the DAC algorithm to fit an aggregated model can be regularized using a Lasso or Ridge penalty to mitigate overfitting. This would be especially useful if some of the features are correlated, which is common in retail settings.

We next show that DAC can consistently identify the aggregation level of each feature and the underlying cluster structure, under sufficiently many observations.

Proposition 1. *DAC outputs a consistent estimate of the aggregation level for each feature and of the underlying cluster structure of the items, that is,*

$$\lim_{m\uparrow\infty}\mathbb{P}\Big[(\hat{D}_n,\hat{D}_s,\hat{D}_c)\neq(D_n,D_s,D_c)\ \textit{or}\ (\hat{\mathcal{C}}_1,\hat{\mathcal{C}}_2,\cdots,\hat{\mathcal{C}}_k)\ \textit{is not a permutation of}\ (\mathcal{C}_1,\mathcal{C}_2,\cdots,\mathcal{C}_k)\Big]=0.$$

As shown in Appendix B.2, the main idea behind the proof of Proposition 1 is to leverage the consistency of the decentralized estimator. The estimated coefficients in the decentralized model will eventually converge to their true values, and hence allow us to accurately learn the aggregation level and the cluster structure. We also want to highlight that the choice of item 1 in the first step of the DAC algorithm is without loss of generality. Choosing another item in this step will also produce a consistent estimator of aggregation levels $(\hat{D}_n,\hat{D}_s,\hat{D}_c)$, cluster structure $(\hat{\mathcal{C}}_1,\hat{\mathcal{C}}_2,...,\hat{\mathcal{C}}_k)$, and feature coefficients $\hat{\beta}$. This follows from the fact that all cluster-level features share the same cluster structure as the items (i.e., $(\mathcal{C}_1,\mathcal{C}_2,...,\mathcal{C}_k)$). Thus, a feature l is identified to be at the cluster level for item i, if and only if it is also identified to be at the cluster level for another item $i'\neq i$.

[1] If we relax Assumption 1, the DAC can easily be adapted to run $O(n^2)$ hypothesis tests instead of $O(n)$.

4. Theoretical Properties of DAC

Since the decentralized model also produces a consistent estimator, the following question arises: What is the benefit of performing the pairwise tests and the clustering algorithm relative to the decentralized model? We address this question from three perspectives: (a) Analytical comparison between the aggregated and decentralized models, which highlights the value of data aggregation and cluster structure; (b) Simulation studies of DAC versus several benchmarks, which show that DAC can successfully identify and leverage the data aggregation and cluster structure; and (c) Implementation of DAC using retail data, which showcases the practical value of DAC in improving demand prediction accuracy. In this section, we examine the value of data aggregation and cluster structure from a theoretical perspective by showing several benefits of the aggregated model relative to the decentralized model.

To convey the benefits of DAC, we first observe that if the true data generating process has different aggregation levels for different features, though flexible, the decentralized model assumes an overly complex model and, hence, will be prone to over-fitting. To formalize this intuition, we leverage the asymptotic normality property of the MLE. Specifically, we denote $\mathcal{I}(\beta)$ as the Fisher's information matrix, which is the Hessian of the log-likelihood evaluated at the true coefficients:

$$\mathcal{I}(\beta) := \text{Hess}\left[\sum_{i=1}^{n}\sum_{j=1}^{m}\log\mathcal{L}(\beta|Y_{i,j}, X_{i,j}^1, \ldots, X_{i,j}^d)\right],$$

where $\beta \in \mathbb{R}^{d_x}$ is the true coefficient vector associated with the true data aggregation and cluster structure. Similarly, we denote $\mathcal{I}_i(\beta_i)$ as the Fisher's information matrix, which is the Hessian of the log-likelihood evaluated at the true coefficients for item i:

$$\mathcal{I}_i(\beta_i) := \frac{\partial^2}{\partial\beta_{i,l}\partial\beta_{i,l'}}\left[\sum_{j=1}^{m}\log\mathcal{L}(\beta_i|Y_{i,j}, X_{i,j}^1, \ldots, X_{i,j}^d)\right],$$

where $\beta_i \in \mathbb{R}^d$ is the true coefficient vector associated with item i. Note that to obtain $\mathcal{I}(\beta)$, we need to specify the data aggregation and the cluster structure, which are not necessary to compute $\mathcal{I}_i(\beta_i)$. We are now ready to compare the (asymptotic) variances of the aggregated and decentralized models. We use $\text{Var}(\cdot)$ to denote the variance operator.

Proposition 2. *For the aggregated and decentralized models, the following statements hold:*
(a) $\hat{\beta}$ *and* $\hat{b}$ *converge to the following asymptotic distributions as* $m \to \infty$,

$$\sqrt{m}(\hat{\beta} - \beta) \xrightarrow{d} N(0, \mathcal{I}(\beta)^{-1}),$$
$$\sqrt{m}(\hat{b}_i - \beta_i) \xrightarrow{d} N(0, \mathcal{I}_i(\beta_i)^{-1}), \text{ for } i = 1, 2, \ldots, n,$$

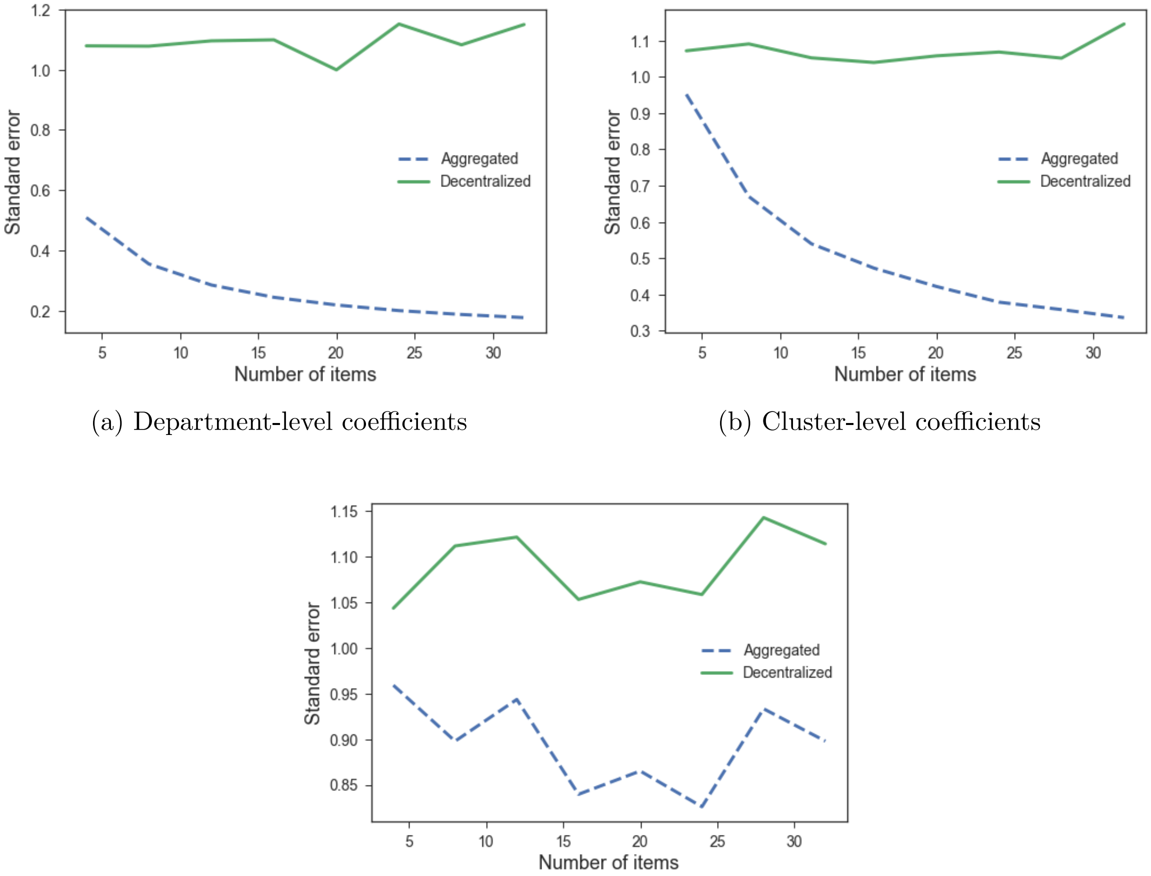
where $\xrightarrow{d}$ *refers to convergence in distribution.*

(b) If $\mathcal{I}_i(\beta_i)$ is diagonal for all i (e.g., in the linear regression model where different columns of X_i are orthogonal with each other for each i), then $\mathcal{I}(\beta)$ is also diagonal. In this case, there exists a constant $\kappa_{i,l} > 0$ for any (i,l), such that

$$\lim_{m\to+\infty} m \cdot n_{i,l} \cdot \mathit{Var}(\hat{\beta}_{i,l}) = \kappa_{i,l}, \ \text{for } i = 1,2,\ldots,n, \ l = 1,2,\ldots,d,$$
$$\lim_{m\to+\infty} m \cdot \mathit{Var}(\hat{b}_{i,l}) = \kappa_{i,l}, \ \text{for } i = 1,2,\ldots,n, \ l = 1,2,\ldots,d.$$

Since $n_{i,l} > 1$ for $l \in D_s \cup D_c$, the aggregated estimation yields a smaller asymptotic variance relative to the decentralized estimation for the coefficients of features at aggregated and cluster levels.

When the number of observations m becomes larger, the variance of the estimated coefficients of both models will shrink to zero. What makes the aggregated model more powerful is its capability to pool the data from different items, thus further reducing the variance of estimation, as shown in Proposition 2. In particular, if a feature is at the aggregated or cluster level, the aggregated model will use at least twice as many observations as the decentralized model to estimate the coefficient of this feature. Hence, the variance will shrink faster especially when n is large. In practice, a typical retail department consists of a large number of items ($n > 100$), so that the aggregated model can be much more efficient than the decentralized model. Proposition 2 further shows that, for the ordinary least squares (OLS) setting, the variances of aggregated and decentralized models can be computed in closed form and, thus, directly comparable. For a general non-linear GLM setting (e.g., logistic regression), however, a closed-form expression of variances cannot be derived. Instead, we will convey the efficiency improvement of the aggregated estimator relative to the decentralized estimator by numerically computing the standard error for the estimated coefficients at different aggregation levels.



(a) Department-level coefficients

(b) Cluster-level coefficients

(c) SKU-level coefficients

Figure 1 Comparison of standard errors for the aggregated and decentralized models

In Figure 1, we consider a simple illustrative example where we fix the number of observations for each item ($m = 50$) and the number of clusters ($k = 4$).[2] For each value of n, we generate 100 independent instances to compute the average standard error of the estimated coefficients for both the aggregated and decentralized estimators (for each type of feature). As n increases, the standard errors of the estimated coefficients for department and cluster levels decrease monotonically for the aggregated model. In contrast, n does not affect the standard errors for the decentralized model. The estimated coefficients improve substantially when n increases. These plots demonstrate the significant efficiency improvement (i.e., reducing variance) of the aggregated model relative to the decentralized model when there are department- and cluster-level features. We next derive probabilistic bounds for the mean squared error under the OLS setting for both the decentralized and aggregated models.

[2] Parameters for Figure 1: $d = 3$ (one feature at each level), β is obtained from a uniform $[-2, 2]$, X from a uniform $[0, 1]$, and $\sigma^2 = 0.1$ (for more details, see Section 5). For each n, we generate the data and estimate both models.

Proposition 3. *Under the OLS setting, we have*

$$\mathbb{P}\left(\frac{||X\beta - X\hat{\beta}||_2^2}{n\times m} \leq \frac{\sigma^2\left(2\sqrt{\gamma d_x}+2\gamma+d_x\right)}{n\times m}\right) \geq 1-\exp(-\gamma)$$

for the decentralized model, and

$$\mathbb{P}\left(\frac{\sum_{i=1}^{n}||X_i b_i - X_i\hat{b}_i||_2^2}{n\times m} \leq \frac{\sigma^2\left(2\sqrt{\gamma\,(nd)}+2\gamma+nd\right)}{n\times m}\right) \geq 1-\exp(-\gamma)$$

for the aggregated model. Furthermore, when

$$d_x \leq nd - 2\left(\sqrt{\gamma nd} + \sqrt{\gamma nd - 2\gamma\sqrt{nd} - \gamma^2}\right)$$

there exists a threshold value Γ *such that,*

$$\mathbb{P}\left(\frac{||X\beta - X\hat{\beta}||_2^2}{n\times m} \leq \Gamma \text{ and } \Gamma \leq \frac{\sum_{i=1}^{n}||X_i b_i - X_i\hat{b}_i||_2^2}{n\times m}\right) \geq 1-\exp(-\gamma), \tag{3}$$

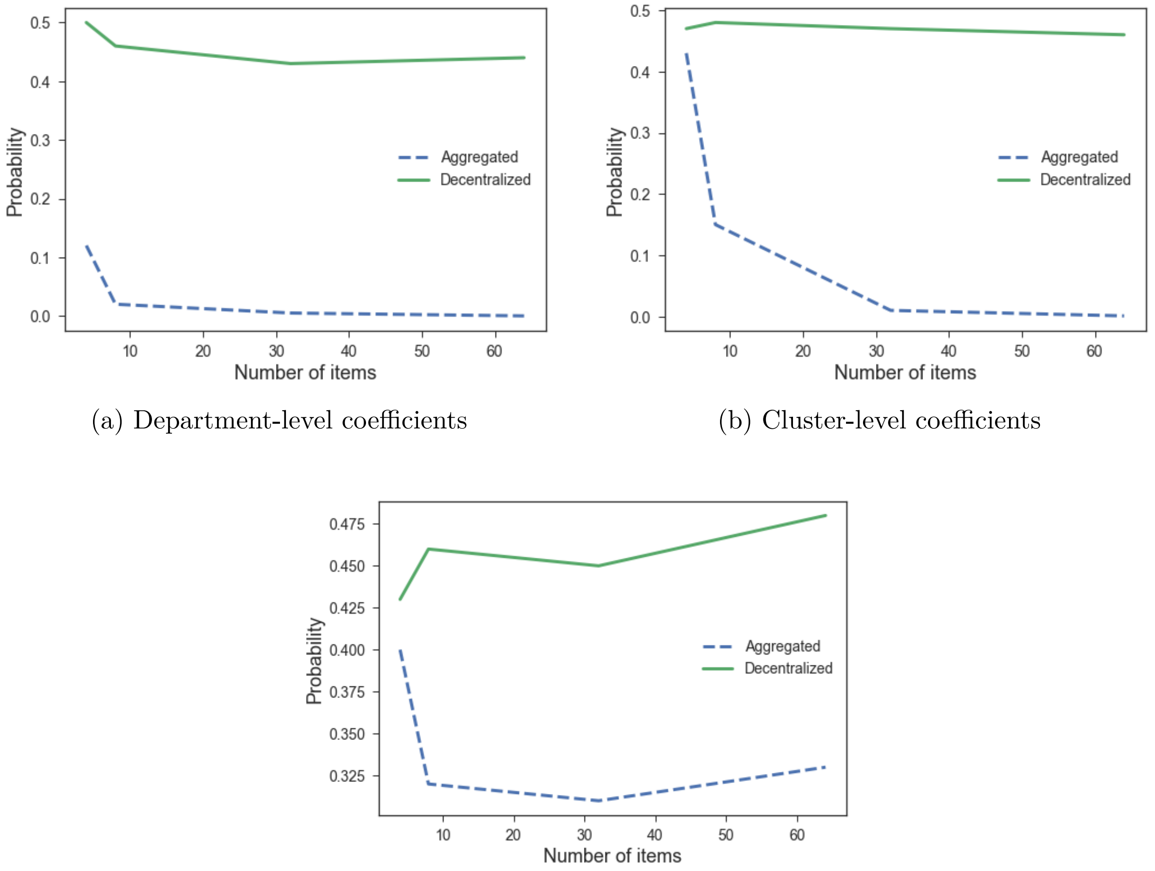
which implies that the aggregated model outperforms the decentralized model with high probability (if we set γ *large enough).*

The parameter γ is a positive constant that captures the desired probability bound, and recall that $d_x = n|D_n| + k|D_c| + |D_s|$ corresponds to the total number of coefficients in our model. Proposition 3 implies that, with high probability, the estimation error of the aggregated model is smaller relative to the decentralized model as long as the number of coefficients is small. Note that condition (3) is well defined only when $\gamma \leq nd - 2\sqrt{nd}$. Table 1 illustrates the result of Proposition 3, that is, how large d_x can be relative to the total number of features nd.

Table 1 Maximum value of d_x

γ	Probability	nd	d_x^*
1	0.632	500	412
2	0.865	1,000	824
5	0.993	3,000	2,514
10	0.999	5,000	4,112

As we can see from Table 1, the upper bound on d_x (denoted by d_x^*) is relatively easy to satisfy. For example, when $nd = 1,000$, it means that for the decentralized model one needs to estimate 1,000 parameters. In this case, the aggregated model will outperform the decentralized model with probability (at least) 0.865 as long as the total number of features is less than 824. Suppose now that the number of parameters in the original model is large (e.g., $nd = 5,000$). As long as there is a non-negligible number of department- and cluster-level features to reduce the number of coefficients to 4,112, the aggregated model will outperform the decentralized model with probability very close to 1. This ultimately illustrates the power of aggregating data and reducing dimensionality to improve prediction accuracy.



(a) Department-level coefficients

(b) Cluster-level coefficients

(c) SKU-level coefficients

Figure 2 Comparison of large estimation error probability for aggregated and decentralized models

For a non-linear GLM model (e.g., logistic regression), since the prediction accuracy does not have a closed form expression, we study the probabilistic bound computationally. For a given coefficient, we compute $\mathbb{P}(|\hat{\beta}_{i,l} - \beta_{i,l}| > \eta)$, which measures the confidence level of the estimate. As we can see from Figure 2, regardless of the aggregation level, the probability that the estimated coefficient is far from the true value is lower for the aggregated model relative to the decentralized model, especially when n is large.[3]

To conclude this section, we remark that all our analysis has focused on comparing the aggregated and decentralized models. We note that the data aggregation level and cluster structure are not

[3] Parameters for Figure 2: $m = 200$, $k = 4$, $d = 3$ (one feature at each level), β is obtained from a uniform $[-2, 2]$, X from a uniform $[0, 1]$, and $\eta = 0.4$. For each n, we generate the data and fit both models. Since we generate 100 independent instances for each n, we count the number of instances where $|\hat{\beta} - \beta| > \eta$ to compute the probability.

known a priori, but are identified via the hypothesis testing and the k-means steps of the DAC. As a result, one could expect additional biases and increased variances for the DAC relative to the aggregated model. Ultimately, one may question whether the value of data aggregation and clustering remains significant for the DAC. Our simulation and real data studies (see Sections 5 and 6) clearly convey that our proposed DAC algorithm efficiently identifies and leverages the data aggregation and cluster structure, and hence substantially improves the out-of-sample prediction accuracy relative to several benchmarks.

5. Simulated Experiments

In this section, we conduct computational experiments using simulated data. We focus on the predictive power of our method and illustrate the improvement in prediction accuracy relative to several benchmarks. We consider two settings under the GLM framework: OLS and logistic regression. The model performance is evaluated using the out-of-sample R^2 for OLS and the area under the curve (AUC) score for logistic regression. We also undertake a comprehensive sensitivity analysis to examine how the different parameters affect the model performance.

5.1. Linear Regression

The data is assumed to be generated from the following linear model:

$$Y_{i,j} = \sum_{l \in D_s} X_{i,j}^l \beta_l^s + \sum_{l \in D_n} X_{i,j}^l \beta_{i,l}^n + \sum_{l \in D_c} X_{i,j}^l \beta_{\mathcal{C}(i),l}^c + \epsilon_{i,j}, \qquad i = 1, \dots, n \text{ and } j = 1, \dots, m,$$

where $\epsilon_{i,j} \sim N(0, \sigma^2)$ are independent and identically distributed random variables. Each data point, $X_{i,j}^l$, is generated randomly from a uniform $[0,1]$ distribution, and each β coefficient is obtained from a uniform $[-2,2]$ distribution. We fix the number of clusters $k = 5$ and vary the parameters $\{n, d, m, \sigma^2, p, q\}$ one at a time. The definition and range of values for these parameters are reported in Table 2. The parameters p and q represent the probability that a given feature is modeled at the department and cluster levels, respectively (different features are drawn independently).

Table 2 Parameters used in Section 5.1

Parameter	Range of values
Number of items (n)	$[10, 150]$
Number of features (d)	$[2, 15]$
Number of observations (m)	$[10, 100]$
Variance of the noise (σ^2)	$[0.05, 0.25]$
Department-level probability (p)	$[0, 2/3]$ or $[0, 1]$
Cluster-level probability (q)	$[0, 2/3]$ or $[0, 1]$

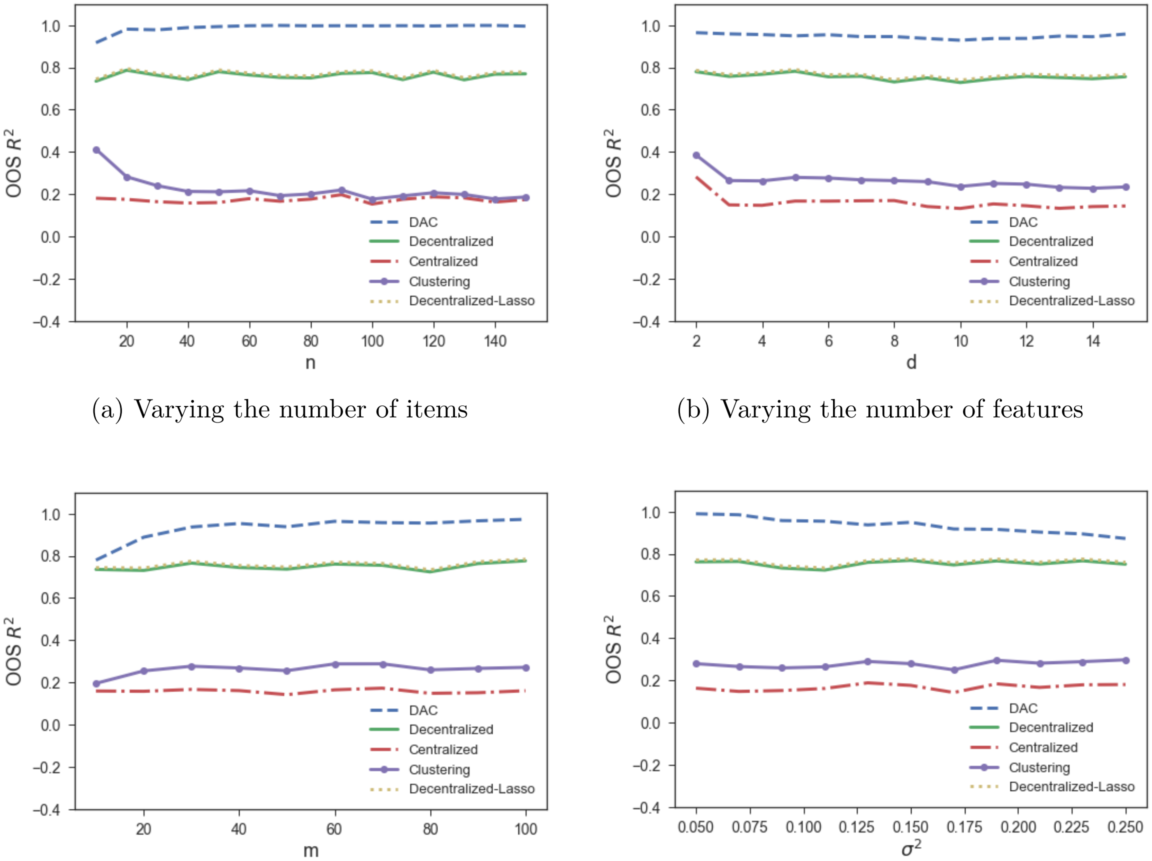
It is important to note that the DAC implementation admits three design parameters: θ, R_U, and R_L in addition to the number of clusters k. These three parameters represent the strictness of our algorithm in determining whether or not a feature should be aggregated. Specifically, θ is the p-value cut-off for statistical significance and is usually 0.05 or 0.01. The parameters R_U and R_L represent the thresholds for the ratio of non-rejected hypotheses. For example, suppose that the percentage of non-rejected hypotheses for feature j is $R_j = 0.3$ (i.e., 30% of the items have statistically close estimated coefficients). Then, we label feature j as a department-level feature if $R_j > R_U$ and a SKU-level feature if $R_j < R_L$. For any intermediate value $R_j \in [R_L, R_U]$, we will label feature j as a cluster-level feature. For instance, one can set $R_L = 0.1$ and $R_U = 0.9$. The parameters θ, R_U, and R_L provide us with flexibility in the tolerance level of the algorithm. When using real data (see Section 6), we will set their values using cross-validation.

To test the performance of our algorithm, we consider the four following benchmarks: Decentralized, Decentralized-Lasso, Centralized, and Clustering. For each instance (i.e., a specific combination of $\{n, d, m, \sigma, p, q\}$), we generate 100 independent trials (datasets) and use 70% as training and 30% as testing. We then report the average out-of-sample R^2. Below is a description of all the methods we consider:

1. **DAC**: We implement our algorithm with $\theta = 0.01$, $R_U = 0.9$, and $R_L = 0.1$.
2. **Decentralized**: We estimate a simple OLS model for each item separately (i.e., n models).
3. **Decentralized-Lasso**: Same as the decentralized method while adding a L_1 regularization term to each OLS model.
4. **Centralized**: This is a naïve OLS model where we assume that, for each feature, all the items have the same coefficient.
5. **Clustering**: We first cluster the items using k-means based on the mean values of the features. We then fit an OLS model for each cluster.

As we can see from Figure 3, our algorithm outperforms all the benchmarks in all settings in terms of out-of-sample R^2. As we increase the number of items or the number of observations, the prediction accuracy of DAC quickly converges to 1, as opposed to the other methods. This clearly demonstrates the power of data aggregation and cluster structure of our algorithm. As expected, a higher σ^2 has a negative impact on the prediction accuracy as it makes structure identification more challenging. Still, our proposed algorithm depicts a substantial advantage relative to the four benchmarks. Finally, varying the number of features does not affect the performance of any of the methods (the performance of each method depends on the proportion of the different feature types and not on the absolute number of features). In addition, we can observe that the DAC has a smoother curve (i.e., fewer "bumps") than the other methods. This implies that our method generates a more stable prediction and, in general, has a smaller variance.

Figure 4 presents the performance of the methods as we vary the structure probability of the features in terms of aggregation level. The first two plots show that if a large proportion of the



(a) Varying the number of items

(b) Varying the number of features

(c) Varying the number of observations

(d) Varying the noise magnitude

Figure 3 Performance comparison of different prediction models (for linear regression)

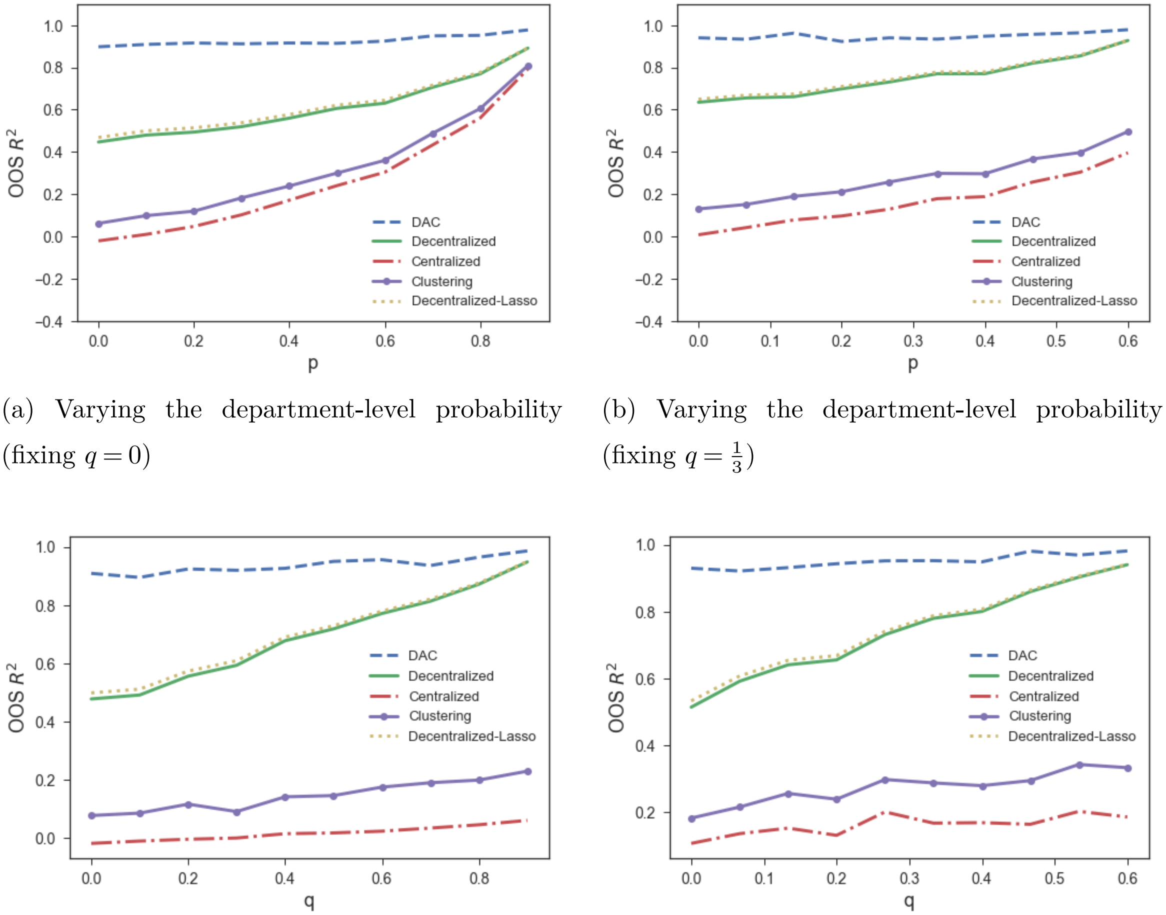
features are at the department-level (i.e., p is close to 1), all five methods perform well (the DAC still performs best in all cases). However, for instances where the structure is more diverse, our algorithm significantly outperforms the four benchmarks. The bottom panels in Figure 4 convey a similar message, except that the Clustering and Centralized methods have a poor performance when the number of department-level features is small.

5.2. Logistic Regression

In this section, we present computational experiments for a classification problem in which the data generating process is the logistic regression model, that is,

$$Y_{i,j} \sim Bernoulli\left(\mu_{i,j}\right), \qquad i=1,\dots,n \text{ and } j=1,\dots,m,$$
$$\mu_{i,j} = \text{logit}\left(\sum_{l\in D_s} X^l_{i,j}\beta^s_l + \sum_{l\in D_n} X^l_{i,j}\beta^n_{i,l} + \sum_{l\in D_c} X^l_{i,j}\beta^c_{\mathcal{C}(i),l}\right),$$

where $\text{logit}(z) := \frac{\exp(z)}{1+\exp(z)}$. We use a similar setting as in Section 5.1. We first generate the data matrix X using a uniform $[0,1]$ distribution and the β coefficients from a uniform $[-2,2]$ distribution. The



(a) Varying the department-level probability (fixing $q = 0$)

(b) Varying the department-level probability (fixing $q = \frac{1}{3}$)

(c) Varying the cluster-level probability (fixing $p = 0$)

(d) Varying the cluster-level probability (fixing $p = \frac{1}{3}$)

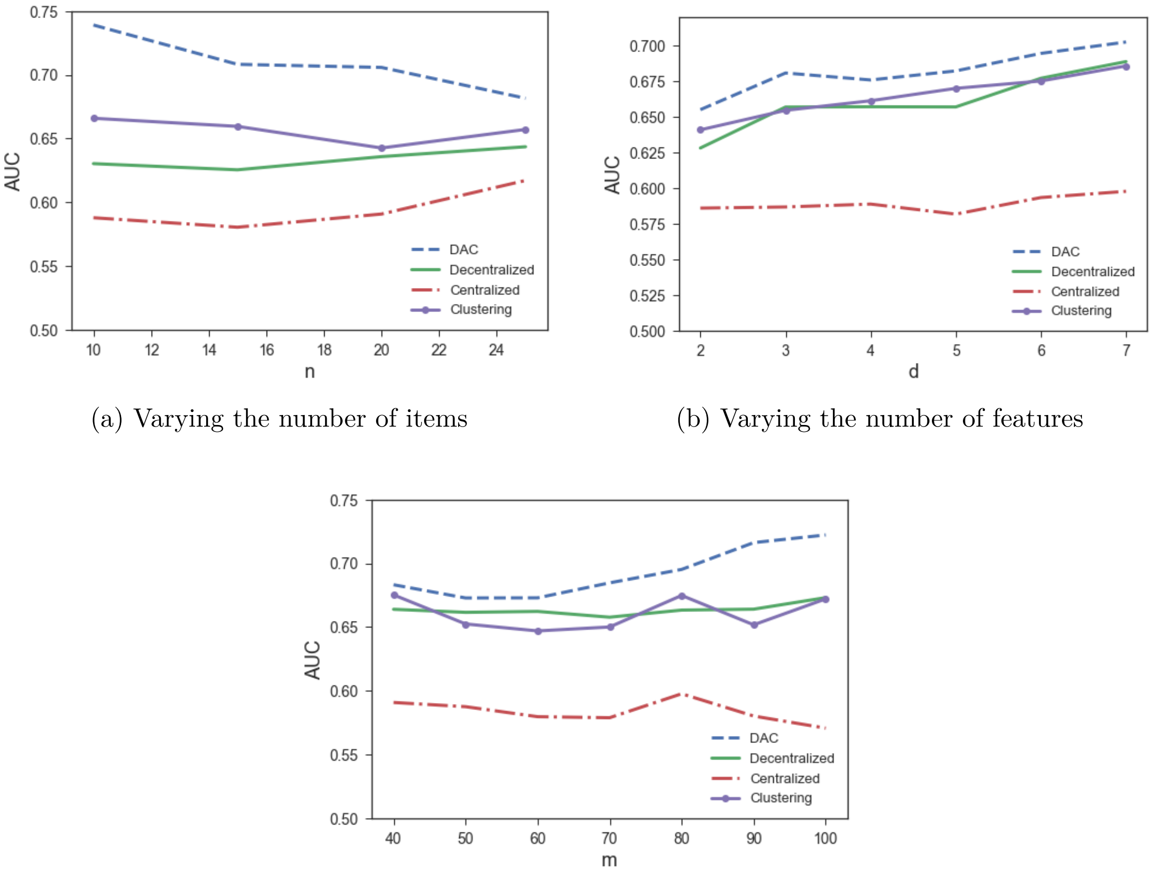
Figure 4 Performance comparison of different prediction models (for linear regression)

outcome variable Y is then generated based on a Bernoulli distribution with parameter $\mu = \text{logit}(X\beta)$. As in the OLS case, we vary one parameter at a time. The ranges of the parameters are summarized in Table 3.

Table 3 Parameters used in Section 5.2

Parameter	Range of values
Number of items (n)	$[10, 25]$
Number of features (d)	$[2, 7]$
Number of observations (m)	$[40, 100]$
Department-level probability (p)	$[0, 2/3]$ or $[0, 1]$
Cluster-level probability (q)	$[0, 2/3]$ or $[0, 1]$

Following several prior studies on binary classification problems, we use the AUC as the metric to evaluate the performance of the different models. AUC is defined as the area under the receiver operating characteristic (ROC) curve (see, e.g., Bradley 1997). It can be interpreted as the probability that a prediction model is correctly ranking a random positive outcome higher than a random negative outcome. We compare our algorithm relative to three benchmarks: Decentralized, Centralized, and Clustering (definitions are similar to Section 5.1).[4] For each instance, we generate 100 independent trials and report the average out-of-sample AUC scores.



(a) Varying the number of items

(b) Varying the number of features

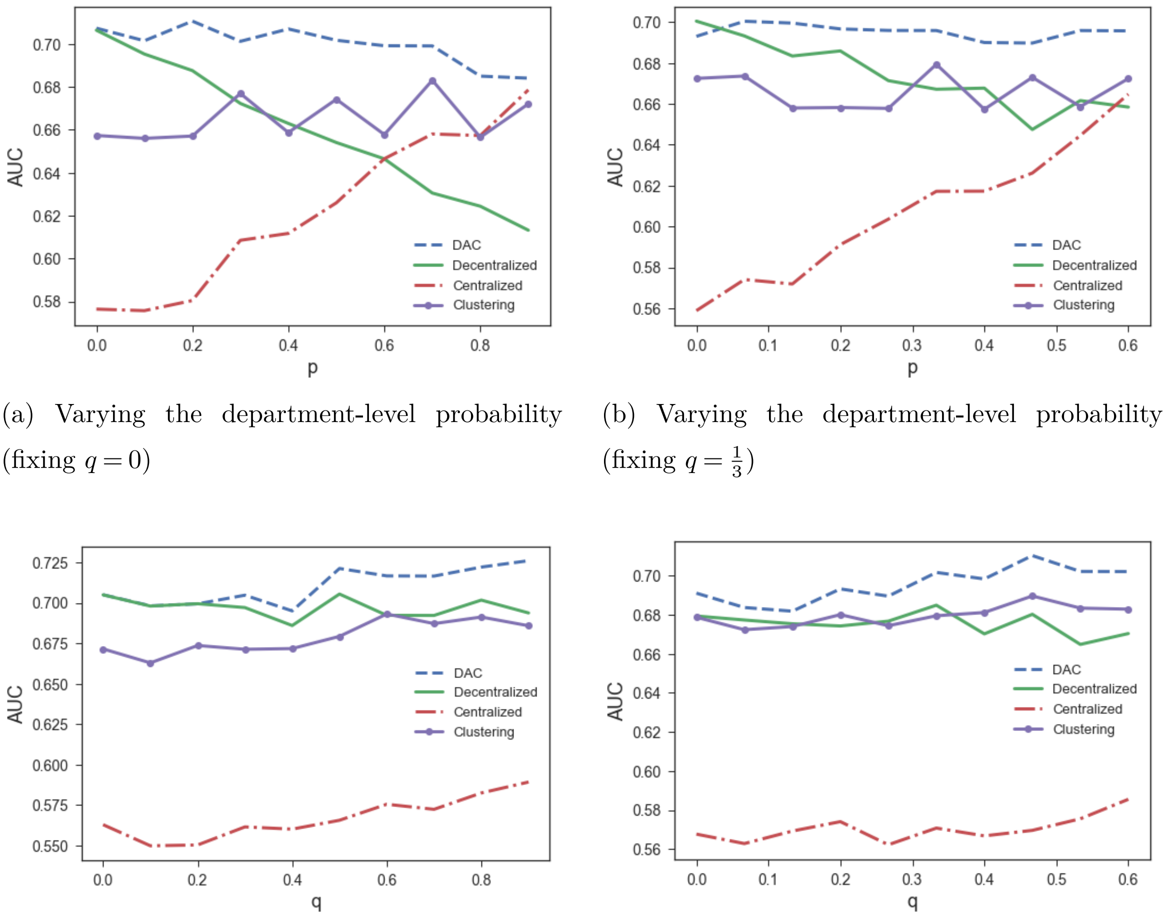
(c) Varying the number of observations

Figure 5 Performance comparison of different prediction models (for logistic regression)

As we can see from Figures 5 and 6, our method outperforms the benchmarks in all cases. Regardless of how we vary $\{n, m, d\}$, the DAC outperforms the three other methods in terms of prediction accuracy. As expected, if the number of department- or cluster- level features are low (i.e., p or q are small), the advantage of the DAC is reduced. However, when at least 30% of the features are at the department or cluster level, our method significantly outperforms the three benchmarks.

To summarize, our simulation studies exhibit a substantial and robust performance improvement for our proposed algorithm relative to several benchmarks (which are commonly used in practice and

[4] Since estimating a decentralized model with L_1 regularization is computationally prohibitive for the logistic regression setting, we only show the performance of the decentralized model without regularization.



(a) Varying the department-level probability (fixing $q = 0$)

(b) Varying the department-level probability (fixing $q = \frac{1}{3}$)

(c) Varying the cluster-level probability (fixing $p = 0$)

(d) Varying the cluster-level probability (fixing $p = \frac{1}{3}$)

Figure 6 Performance comparison of different prediction models (for logistic regression)

in the literature). For both the regression and classification problems, the DAC efficiently aggregates the data and identifies the cluster structure of the items, and thus improves the prediction accuracy. In the next section, we apply the DAC to actual retail data to showcase the benefits in a practical business setting.

6. Applying DAC to Retail Data

In this section, we apply the DAC using a retail dataset from a large global retail firm (we do not reveal the name of the retailer due to a non-disclosure agreement). We first provide a detailed description of the data, and then test the prediction performance of our model relative to several benchmarks. Finally, based on our computational findings, we draw useful managerial insights that can help retailers infer which features should be aggregated in practice.

6.1. Data

We have access to the online sales data from the retailer. The dataset is comprehensive and includes several departments. Specifically, the data records weekly sales information of five departments between November 2013 and October 2016. A typical department comprises of 80-150 SKUs. In addition to the weekly sales information, the dataset also includes weekly price, promotion indicator (whether or not an item was promoted), vendor, and the color of the SKU. Table 4 summarizes the specifics of each department. The size corresponds to the number of items in each department and the number in parenthesis are the standard deviations.

Table 4 Summary statistics of each department

Dept	Size	Observations	Weekly sales	Price	Promo Frequency	Discount rate
1	117	19,064	108.11 (377.45)	42.92 (38.80)	31.3%	4.2%
2	134	20,826	254.23 (517.07)	8.94 (8.67)	7.7%	6.4%
3	113	17,207	200.17 (452.97)	12.63 (7.02)	8.9%	37.7%
4	125	14,457	68.68 (391.52)	97.61 (67.12)	8.5%	1.5%
5	84	12,597	115.62 (337.29)	22.17 (17.37)	6.3%	6.7%

As we can see from Table 4, each department has a large number of observations (observations are at the week-SKU level). In addition, we have a great variation in terms of weekly sales, price, promotion frequency, and discount rate across the different departments. Table 5 provides a brief description of the different fields in our dataset. The effective weekly price is computed as the total weekly revenue divided by the total weekly sales. Functionality is a segmentation hierarchy used by the firm to classify several SKUs from the same department into sub-categories.

Table 5 Fields in our dataset (observations are aggregated at the week-SKU level)

Fields	Description
SKU ID	Unique SKU ID
Week	Week index
Year	Year index
Units	Total weekly sales of a specific SKU
Price	Effective weekly price of a specific SKU
PromoFlag	Whether there was a promotion during that week
Functionality	Class index of a specific SKU
Color	Color of a specific SKU
Vendor	Vendor of a specific SKU

Based on the features available in our data, we estimate the following baseline regression model:

$$\begin{aligned} Y_{i,t} = {} & \beta^i_{\text{Trend}} \cdot T_{i,t} + \beta^i_0 \cdot p_{i,t} + \beta^i_1 \cdot \text{PromoFlag}_{i,t} + \beta^i_2 \cdot \text{Fatigue}_{i,t} + \beta^i_3 \cdot \text{Seasonality}_{i,t} + \\ & + \beta^i_4 \cdot \text{Functionality}_{i,t} + \beta^i_5 \cdot \text{Color}_{i,t} + \beta^i_6 \cdot \text{Vendor}_{i,t} + \epsilon_{i,t}, \end{aligned} \tag{4}$$

Equation (4) includes the following features:

1. $Y_{i,t}$: total weekly sales of item i in week t (our dependent variable),

2. $T_{i,t}$: trend variable of item i. We normalize the year so that $T_i = 0, 1, 2, 3$,

3. $p_{i,t}$: effective price of item i in week t,

4. $\text{PromoFlag}_{i,t}$: binary variable indicating whether there is a promotion for item i in week t,

5. $\text{Fatigue}_{i,t}$: number of weeks since the most recent promotion for item i. If there is no previous promotion, $Fatigue_{i,t} = 0$. This feature allows us to capture the post-promotion dip effect.

6. Seasonality: categorical variable that measures the weekly or monthly effect on sales. We use the one-hot encoding in our model.[5]

7. $\epsilon_{i,t}$: additive unobserved error.

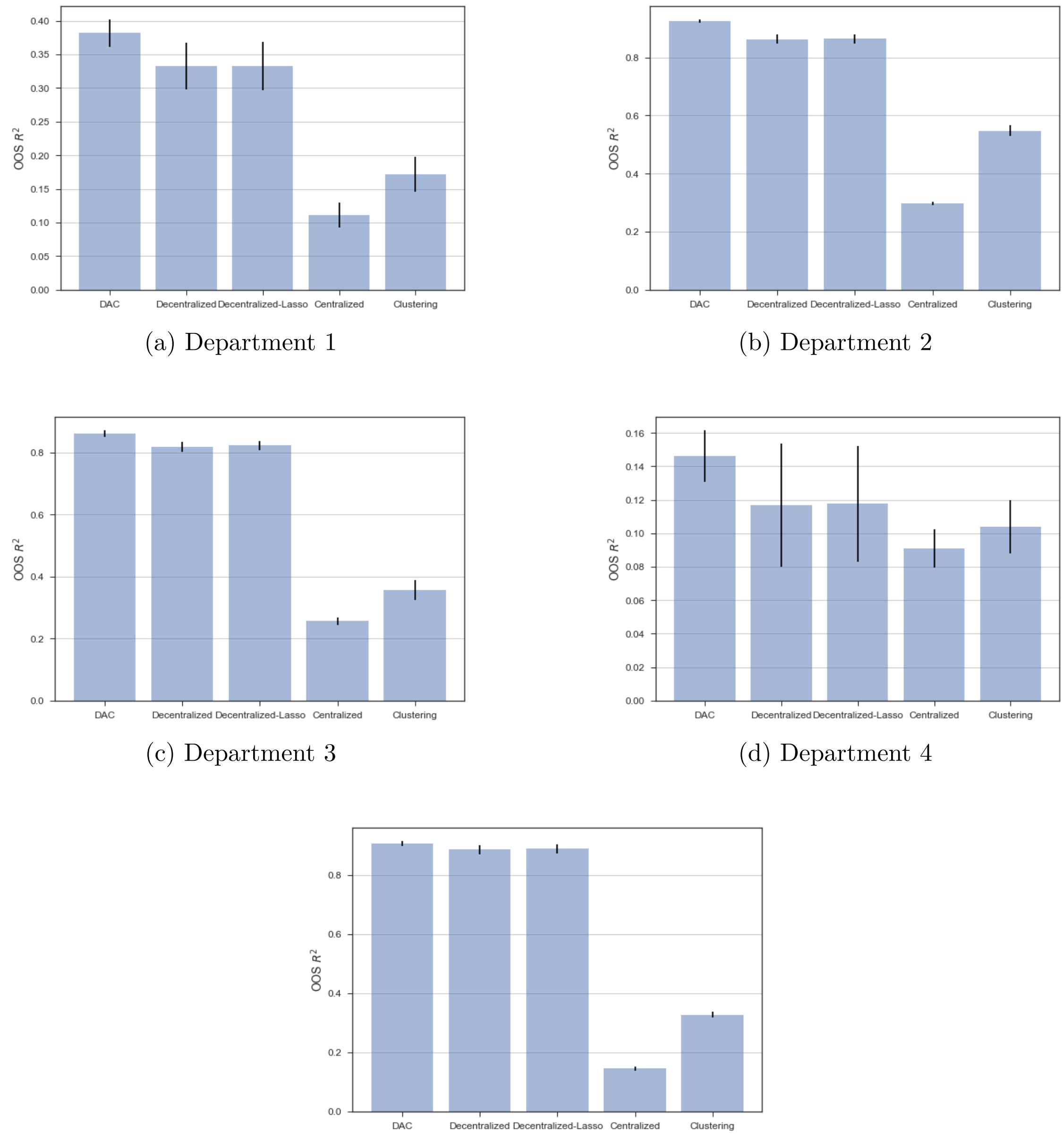
The remaining three variables are categorical variables indicating the web class index (Functionality), Color, and Vendor of the SKU, respectively.

6.2. Prediction Performance

As in Section 5, we investigate the performance of our algorithm relative to the same four benchmarks (Decentralized, Decentralized-Lasso, Centralized, and Clustering). It is important to mention that when implementing our algorithm using real data, we need to slightly adapt the clustering step. To make sure that we output a single cluster structure, we first collect the estimated coefficients from all cluster-level features, and then fit a multi-dimensional k-means model. To avoid overfitting, we also include a L_1-regularization term in the last step of the aggregated estimation.

Since DAC(k, θ, R_U, R_L) has four design parameters, we use cross-validation for model selection. For each department, we first randomly split the data into training (70%) and testing (30%). We assume that each design parameter lies within a pre-specified range: $k \in \{3, 4, \ldots, 10\}$, $\theta \in \{0.01, 0.05, 0.1\}$, $R_U \in \{0.7, 0.8, 0.9\}$, and $R_L \in \{0.1, 0.2, 0.3\}$. For each combination of design parameters, we perform a five-fold cross-validation by fitting the model on 80% of the training data and compute the R^2 based on the remaining 20% of the data. This procedure is repeated five times for each different parameter combination, and we compute the average R^2 over the five folds. We next select the best model based on the average cross-validation performance. Finally, the out-of-sample R^2 is computed using the test set. Furthermore, since the train/cross-validation/test split is done randomly, we conduct 100 independent trials and report both the mean and standard deviation of the out-of-sample R^2. The following bar charts summarize the prediction performance of the DAC.

[5] https://en.wikipedia.org/wiki/One-hot



(a) Department 1

(b) Department 2

(c) Department 3

(d) Department 4

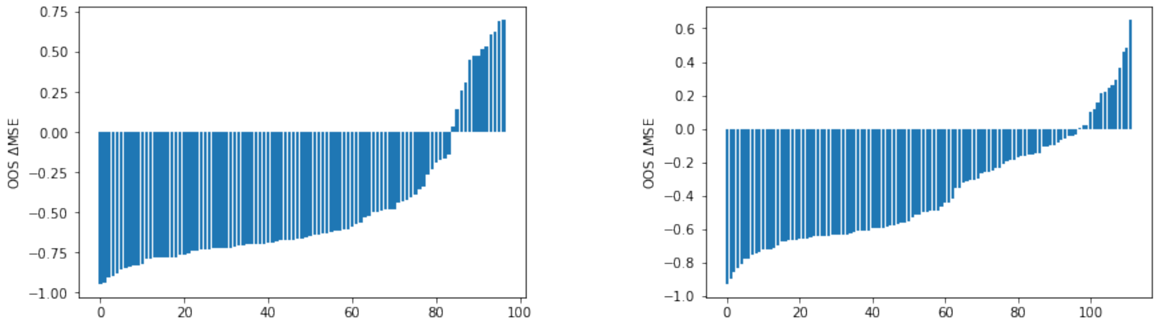
(e) Department 5

Figure 7 Performance comparison using real data

In Figure 7, each bar represents the average out-of-sample performance and the length of the vertical line corresponds to the standard deviation across 100 independent trials. As we can see, for all five departments, our algorithm not only achieves a better average prediction performance but also has a smaller variance. This shows that our algorithm is robust to different train/test splits, which is very desirable in practice. In addition, our method outperforms all four benchmarks, regardless of data quality. More precisely, Department 2 seems to have a high-quality data, whereas the data for Department 4 seems to be of lower quality (the number of observations per SKU is the lowest for

Department 4 and the variability is high). Irrespective of the data quality, the DAC yields a clear improvement in prediction accuracy.

We next present a per-item comparison between Departments 1 and 2. Specifically, we compare the performance of our algorithm relative to the second best method, that is, Decentralized-Lasso for each item in the department. To mitigate the effect of outliers, we remove the bottom and top 5% SKUs in terms of mean squared error (MSE) for both methods.



(a) Department 1

(b) Department 2

Figure 8 Per-item prediction accuracy comparison

The y-axis in Figure 8 is the relative out-of-sample MSE, that is, $\frac{MSE_i(DAC)-MSE_i(Dece)}{MSE_i(Dece)}$, which measures the percentage improvement of DAC relative to Decentralized-Lasso for each item i in the department. Note that we report the per-item MSE instead of the per-item R^2, given that the per-item R^2 is not a well-defined metric. Our algorithm improves the demand prediction for more than 70% of the items in both departments. More generally, when applying the DAC to all five departments, we obtain an improvement for 412 out of the 573 SKUs (i.e., 71.9% of the items). Even though the Decentralized-Lasso method yields a good performance in many cases (recall that it avoids overfitting), its performance can be limited when the number of observations per item is small and the number of features is large. On the other hand, our algorithm can adaptively identify the right level of aggregation for each feature, thus alleviating the over-fitting issue and increasing prediction accuracy.

6.3. Managerial Insights

So far, we focused on the prediction performance of the DAC. We next apply the DAC to our dataset and examine the estimation output. Our goal is to draw managerial insights on the hierarchical structure of the features among the different SKUs. We next summarize our findings.

- The DAC can significantly reduce the model dimension. In Table 6, we report the number of estimated coefficients for the Decentralized and DAC methods across all five departments. For

Departments 1 and 2, the number of estimated coefficients reduces by 40%, and for Departments 3–5, the reduction exceeds 50%. The results in Table 6 confirm that shared coefficients do occur in practice, and that data aggregation can play an important role in correctly identifying the aggregation structure.

Table 6 Number of estimated coefficients

Department	Decentralized	DAC
1	7,488	4,360
2	6,298	3,912
3	2,712	1,080
4	3,000	892
5	1,848	729

- Practitioners often argue that seasonality features should be aggregated at the department level for demand prediction (e.g., Cohen et al. 2017b, Vakhutinsky et al. 2018). Using our retail dataset, we discover that this is indeed the case. If we model the seasonality at the month level (i.e., we use 12 dummy variables for each calendar month), we find that at least 10 out of the 12 variables should be estimated at the department level, for each of the five departments. If we instead model the seasonality at the week level (i.e., we use 52 dummy variables for each week of the year), we find that over 90% of the variables should be estimated at the department level. Thus, our findings validate and refine a well-known business practice.
- The price feature is unarguably one of the most important features for demand prediction in retail. According to our estimation results, for departments with a heterogeneous item collection (Departments 1, 2, and 4), we obtain a distinct coefficient for the price feature, implying that the price coefficient should be estimated at the SKU level. On the other hand, for departments in which the product discrepancy is low (Departments 3 and 5), the DAC infers that the price coefficient should be estimated at the cluster level.
- We find that the fatigue and promotion features should be estimated at the department level, for all five departments. This is an interesting insight that can guide retailers when deciding their promotion strategy.
- We obtain that all vendor and color dummy variables should be estimated at the SKU level. This is not surprising given that most vendor-color combinations are unique for a specific SKU.
- The functionality dummy variables have different aggregation levels and cluster structures. Interestingly, most cluster-level features come from this variable. One possible explanation is that the functionality feature is obtained based on the hierarchy structure used by the company. Thus, SKUs with similar characteristics are usually labeled under the same functionality, making the cluster structure more prominent for functionality features. Retailers can use such results to potentially revise and improve their hierarchical structure and product segmentation.

7. Conclusion

Demand prediction (or sales forecast) is an important task faced by most retailers. Improving the prediction accuracy and drawing insights on data aggregation can significantly impact retailers' decisions and profits. When designing and estimating predictive models, retailers need to decide the aggregation level of each model feature. Some features may be estimated at the SKU level, others at the department level, and the rest at a cluster level. Traditionally, this problem was addressed by trial-and-error or by relying on past experience. It is common to see data scientists testing a multitude of model specifications until they find the best aggregation level for each feature. Such an approach can be tedious and does not scale for models with a large number of features. The goal of this paper is to develop an efficient method to simultaneously determine (i) the right aggregation level of each feature, (ii) the underlying cluster structure, and (iii) the estimated coefficients.

We propose a method referred to as the Data Aggregation with Clustering (DAC) algorithm. The DAC can determine the right aggregation level and identify the cluster structure of the items. This method is tractable even when the data dimensionality is high and can significantly improve the efficiency in estimating the model coefficients. We first derive several analytical results to demonstrate the benefit of aggregating similar coefficients. Specifically, we show that the DAC yields a consistent estimate along with improved asymptotic properties relative to the traditional OLS method. We then go beyond the theory and implement the DAC using a large retail dataset. In all our computational tests, we observe that the DAC significantly improves the prediction accuracy relative to several benchmarks. Finally, we convey that our method can help retailers discover useful insights from their data.

References

Baardman L, Levin I, Perakis G, Singhvi D (2017) Leveraging comparables for new product sales forecasting, available at SSRN 3086237.

Bernstein F, Modaresi S, Sauré D (2018) A dynamic clustering approach to data-driven assortment personalization. *Management Science* .

Bertsimas D, Kallus N (2014) From predictive to prescriptive analytics, arXiv preprint arXiv:1402.5481.

Bradley AP (1997) The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition* 30(7):1145–1159.

Caro F, Gallien J (2010) Inventory management of a fast-fashion retail network. *Operations Research* 58(2):257–273.

Cohen M, Kalas J, Perakis G (2017a) Optimizing promotions for multiple items in supermarkets, available at SSRN 3061451.

Cohen MA, Lee HL (2019) Designing the right global supply chain network .

Cohen MC, Leung NHZ, Panchamgam K, Perakis G, Smith A (2017b) The impact of linear optimization on promotion planning. *Operations Research* 65(2):446–468.

Cooper LG, Baron P, Levy W, Swisher M, Gogos P (1999) Promocast™: A new forecasting method for promotion planning. *Marketing Science* 18(3):301–316.

Donti P, Amos B, Kolter JZ (2017) Task-based end-to-end model learning in stochastic optimization. *Advances in Neural Information Processing Systems*, 5484–5494.

Elmachtoub AN, Grigas P (2017) Smart" predict, then optimize", arXiv preprint arXiv:1710.08005.

Fahrmeir L, Kaufmann H, et al. (1985) Consistency and asymptotic normality of the maximum likelihood estimator in generalized linear models. *The Annals of Statistics* 13(1):342–368.

Feng Q, Shanthikumar JG (2018) How research in production and operations management may evolve in the era of big data. *Production and Operations Management* 27(9):1670–1684.

Fildes R, Goodwin P, Önkal D (2019) Use and misuse of information in supply chain forecasting of promotion effects. *International Journal of Forecasting* 35(1):144–156.

Foekens EW, Leeflang PS, Wittink DR (1998) Varying parameter models to accommodate dynamic promotion effects. *Journal of Econometrics* 89(1-2):249–268.

Greene WH (2003) *Econometric analysis* (Pearson Education India).

Hu K, Acimovic J, Erize F, Thomas DJ, Van Mieghem JA (2017) Forecasting product life cycle curves: Practical approach and empirical analysis, manufacturing & Service Operations Management.

Huang T, Fildes R, Soopramanien D (2014) The value of competitive information in forecasting fmcg retail product sales and the variable selection problem. *European Journal of Operational Research* 237(2):738–748.

Huang T, Fildes R, Soopramanien D (2019) Forecasting retailer product sales in the presence of structural change, european Journal of Operational Research.

Jagabathula S, Subramanian L, Venkataraman A (2018) A model-based embedding technique for segmenting customers. *Operations Research* 66(5):1247–1267.

Kao Yh, Roy BV, Yan X (2009) Directed regression. *Advances in Neural Information Processing Systems*, 889–897.

Kesavan S, Gaur V, Raman A (2010) Do inventory and gross margin data improve sales forecasts for us public retailers? *Management Science* 56(9):1519–1533.

Kök AG, Fisher ML (2007) Demand estimation and assortment optimization under substitution: Methodology and application. *Operations Research* 55(6):1001–1021.

Laurent B, Massart P (2000) Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics* 1302–1338.

Lin M, Lucas Jr HC, Shmueli G (2013) Research commentary—too big to fail: large samples and the p-value problem. *Information Systems Research* 24(4):906–917.

Liu S, He L, Shen M (2019) On-time last mile delivery: Order assignment with travel time predictors .

Macé S, Neslin SA (2004) The determinants of pre-and postpromotion dips in sales of frequently purchased goods. *Journal of Marketing Research* 41(3):339–350.

MacQueen J, et al. (1967) Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1(14), 281–297 (Oakland, CA, USA).

Marroquin JL, Girosi F (1993) Some extensions of the k-means algorithm for image segmentation and pattern classification. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB.

McCullagh P (2019) *Generalized linear models* (Routledge).

Pekgün P, Menich RP, Acharya S, Finch PG, Deschamps F, Mallery K, Sistine JV, Christianson K, Fuller J (2013) Carlson rezidor hotel group maximizes revenue through improved demand management and price optimization. *Interfaces* 43(1):21–36.

Vakhutinsky A, Mihic K, Wu SM (2018) A prescriptive analytics approach to markdown pricing for an e-commerce retailer, URL `http://dx.doi.org/10.13140/RG.2.2.35292.69767`, working paper.

Van Heerde HJ, Leeflang PS, Wittink DR (2000) The estimation of pre-and postpromotion dips with store-level scanner data. *Journal of Marketing Research* 37(3):383–395.

Wang H, Song M (2011) Ckmeans. 1d. dp: optimal k-means clustering in one dimension by dynamic programming. *The R journal* 3(2):29.

Appendix A: Two Potential Methods

In this section, we elaborate on the key difficulties of two intuitive methods—*constrained MLE* and *iterative optimization*—in estimating Equation (1).

A.1. Constrained MLE

Constrained MLE adapts the standard MLE approach to account for the equality constraints $\beta_{i,l} = \beta_{j,l}$, for all i, j if l is an aggregated-level feature; and $\beta_{i,l} = \beta_{j,l}$, for all i, j in the same cluster if l is a cluster-level feature. This method, however, is challenging to implement. We next use a simple example to illustrate the main difficulty. Consider a linear regression model with two items (each with one feature) but the level of aggregation is unknown. The data matrix can be written as follows:

$$X = \begin{bmatrix} X_1^1 & 0 & X_1^1 \\ X_2^1 & 0 & X_2^1 \\ ... & ... & ... \\ X_m^1 & 0 & X_m^1 \\ 0 & X_1^2 & X_1^2 \\ 0 & X_2^2 & X_2^2 \\ ... & ... & ... \\ 0 & X_m^2 & X_m^2 \end{bmatrix}.$$

The data matrix X enumerates all possible column combinations (three in this case), which represent all possible options for feature aggregation. If the feature is at the SKU level, then only the first two columns will have non-zero coefficients. On the other hand, if the feature is at the department level, only the last column will have a non-zero coefficient. We denote the vector of parameters as $\beta = (\beta_1, \beta_2, \beta_{12})$. The aggregation level inference problem can be formulated as the following constrained MLE problem (for linear regression, MLE is equivalent to minimizing the squared error):

$$\begin{aligned} \min_{\beta} \quad & ||X\beta - y||^2 \\ \text{s.t.} \quad & \beta_1 \beta_{12} = 0, \\ & \beta_2 \beta_{12} = 0. \end{aligned}$$

Applying the KKT condition, we obtain the following matrix-form optimality condition:

$$2X^T X\beta - 2X^T y + B^T \lambda = 0,$$

where

$$B = \begin{bmatrix} \beta_{12} & 0 & \beta_1 \\ 0 & \beta_{12} & \beta_2 \end{bmatrix}.$$

The complete KKT condition is then given by:

$$\begin{bmatrix} 2X^T X & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \beta \\ \lambda \end{bmatrix} = \begin{bmatrix} 2X^T y \\ 0 \end{bmatrix}$$

which is a non-convex quadratic equation. Thus, the traditional approach to solve a constrained linear regression does not apply in this case. In addition, due to the non-convexity, it is not easy to solve the above optimization problem when the number of items and/or the number of features become large. For a non-linear GLM model, the constrained MLE approach is even more challenging. As a result, the constrained MLE does not seem to be an efficient method to solve our problem.

A.2. Iterative Optimization

The iterative optimization approach was proposed by Baardman et al. (2017). We use the same example as the constrained MLE (i.e., a linear model with SKU- and department-level features only), to illustrate the difficulty of applying the iterative optimization procedure. We formulate the estimation problem as the following optimization program:

$$\begin{aligned} \min_{\delta,\beta_n,\beta_0} \quad & \sum_{j=1}^{m}\sum_{i=1}^{n}\left(Y_{i,j}-\sum_{l=1}^{d}\left[X_{i,j}^{l}\beta_{i,l}\delta_l+X_{i,j}^{l}\beta_{0,l}(1-\delta_l)\right]\right)^2 \\ \text{s.t.} \quad & \beta_{i,l}\in\mathbb{R}^d \;\; i=1,2,\ldots,n,\;\; l=1,2,\ldots,d, \\ & \delta_l\in\{0,1\} \;\; l=1,2,\ldots,d, \end{aligned} \tag{5}$$

where i is the observation index, j the item index, and l the feature index. $\beta_0=(\beta_{0,1},\beta_{0,2},\ldots,\beta_{0,d})'$ (resp. $\beta_j=(\beta_{j,1},\beta_{j,2},\ldots,\beta_{j,d})'$) are the department-level (resp. SKU-level) coefficients for the features. This iterative optimization formulation is a quadratic mixed-integer program. Note that the model is simplified since it does not include cluster-level coefficients.

One can solve the optimization problem in (5) by using the following iterative algorithm:

1. Randomly initialize $\hat{\delta}^0\in\{0,1\}^d$ and solve for $\hat{\beta}^0=\{\hat{\beta}^0_{i,l}: i=1,2,\ldots,l=1,2,\ldots,d\}$ by fixing $\hat{\delta}=\hat{\delta}^0$.
2. Starting from iteration $t=1$, first solve for $\hat{\delta}^t$ by fixing $\beta=\hat{\beta}^{t-1}$; then solve for $\hat{\beta}^t$ by fixing $\delta=\hat{\delta}^t$.
3. Terminate Step 2 when $\hat{\delta}^t=\hat{\delta}^{t-1}$. The output is $(\hat{\beta}^t,\hat{\delta}^t)$.

The difficulty of the iterative optimization procedure is that without any prior information on β, the generation of coefficient vector is random. Therefore, if the generated coefficients are far from their true values, the procedure can converge to a local optimal solution, and hence potentially yield a large estimation error. Furthermore, it is not clear how to incorporate the (unknown) cluster structure into the iterative optimization procedure.

Appendix B: Proofs of Statements

B.1. Proof of Lemma 1

The proof follows from a standard result in statistics stating that the maximum likelihood estimator (MLE) is consistent under some regularity conditions which are satisfied by a generalized linear model. See, for example, Fahrmeir et al. (1985) and McCullagh (2019).

B.2. Proof of Proposition 1

We analyze each aggregation level separately. The results build on Lemma 1 stating that all $\hat{b}_{i,j}$ are consistent.

- Case 1: Feature l is at the SKU level.

In this case, $b_{1,l}\neq b_{k,l}$ for all k in the decentralized model. As a result, based on Lin et al. (2013), for any $\eta>0$, we have that the $p-$value for the hypothesis $H_0: b_{1,l}=b_{k,l}$ satisfies that

$$\lim_{m\to\infty} p\text{-value} = \lim_{m\to\infty} Pr(|\hat{b}_{1,l}-\hat{b}_{k,l}|<\eta)=0. \tag{6}$$

Therefore, the p-value converges to 0 as $m\to\infty$. Alternatively if $l\notin D_n$, there exists an item k', such that

$$\lim_{m\to\infty} p\text{-value} = \lim_{m\to\infty} \mathbb{P}(|\hat{b}_{1,l}-\hat{b}_{k',l}|\geq\eta)=1. \tag{7}$$

Combining inequalities (6) and (7) imply that the probability that we misclassify feature l (in terms of whether or not feature l is at the SKU level) converges to 0 as $m \to \infty$, that is, the DAC consistently identifies whether or not feature l is at the SKU level.

- Case 2: Feature l is at the department level.

In this case, $b_{1,l} = b_{k,l}$ in the decentralized model. Again according to Lin et al. (2013), for any $\eta > 0$, we have,

$$\lim_{m\to\infty} p\text{-value} = \lim_{m\to\infty} Pr(|\hat{b}_{1,j} - \hat{b}_{k,j}| < \eta) = 1. \tag{8}$$

Therefore, the p-value converges to 1 as $m \to \infty$. Alternatively if $l \notin D_s$, there exists an item k', such that

$$\lim_{m\to\infty} p\text{-value} = \lim_{m\to\infty} \mathbb{P}(|\hat{b}_{1,l} - \hat{b}_{k',l}| \geq \eta) = 0. \tag{9}$$

Combining inequalities (8) and (9) imply that the probability that we misclassify feature l (in terms of whether or not feature l is at the department level) converges to 0 as $m \to \infty$, that is, the DAC consistently identifies whether or not feature l is at the department level.

- Case 3: Feature l is at the cluster level.

As in the previous cases, we know that as $m \to \infty$, all the coefficients center around their true values with an arbitrarily high probability. Given the number of clusters k, the task is to partition $\{\hat{b}_{i,l} : i = 1, 2, , \cdots, n\}$ into k groups such that the sum of squared Euclidean distances to each group mean is minimized. In general, the high-dimensional k-means algorithm is NP-hard. However, for this specific one-dimensional k-means problem, there exists a dynamic programming algorithm, with a polynomial time complexity, that finds the optimal solution (Wang and Song 2011). This implies that the DAC consistently identifies whether or not feature l is at the cluster level, and hence concludes the proof. □

B.3. Proof of Proposition 2

(a) The result follows from Theorem 3 in Fahrmeir et al. (1985).

(b) If $\mathcal{I}_i(\beta_i)$ is diagonal for each i, it means that

$$\frac{\partial^2}{\partial\beta_{i,l}\partial\beta_{i,l'}}\left[\sum_{j=1}^{m} \log \mathcal{L}(\beta_i|Y_{i,j}, X_{i,j}^1, \ldots, X_{i,j}^d)\right] = 0, \qquad \forall l \neq l'.$$

This implies that for any two different coefficients, the second-order derivative evaluated at the true coefficients is equal to 0. Therefore, for any off-diagonal entries of $\mathcal{I}(\beta)$, we have

$$\begin{aligned}\mathcal{I}(\beta)_{u,v} &= \frac{\partial^2}{\partial\beta_u\partial\beta_v}\left[\sum_{i=1}^{n}\sum_{j=1}^{m} \log \mathcal{L}(\beta|Y_{i,j}, X_{i,j}^1, \ldots, X_{i,j}^d)\right] \\ &= \sum_{i=1}^{n}\left[\frac{\partial^2}{\partial\beta_u\partial\beta_v}\sum_{j=1}^{m} \log \mathcal{L}(\beta|Y_{i,j}, X_{i,j}^1, \ldots, X_{i,j}^d)\right] = 0, \qquad \forall u \neq v.\end{aligned}$$

To analyze the diagonal terms of the information matrix, we assume that

$$\lim_{m\to+\infty} \frac{1}{m}\frac{\partial^2}{\partial\beta_{i,l}^2}\left[\sum_{j=1}^{m} \log \mathcal{L}(\beta_i|Y_{i,j}, X_{i,j}^1, \ldots, X_{i,j}^d)\right] = \frac{1}{\kappa_{i,l}},$$

where $\kappa_{i,l} > 0$ is a constant for some item i and feature l. We then have

$$\lim_{m\to+\infty} m \cdot \mathrm{Var}(\hat{b}_{i,l}) = \lim_{m\to+\infty} \frac{m}{\frac{\partial^2}{\partial\beta_{i,l}^2}\left[\sum_{j=1}^{m} \log \mathcal{L}(\beta_i|Y_{i,j}, X_{i,j}^1, \ldots, X_{i,j}^d)\right]} = \kappa_{i,l}.$$

One can now derive the asymptotic variance of the coefficient under aggregated estimation:

$$\begin{aligned}\lim_{m\to+\infty} m\cdot \mathrm{Var}(\hat{\beta}_{i,l}) &= \lim_{m\to+\infty} \frac{m}{\frac{\partial^2}{\partial\beta_{i,l}^2}\left[\sum_{i=1}^n\sum_{j=1}^m \log\mathcal{L}(\beta_i|Y_{i,j},X_{i,j}^1,\dots,X_{i,j}^d)\right]}\\ &= \lim_{m\to+\infty} \frac{m}{n_{i,l}\cdot\frac{\partial^2}{\partial\beta_{i,l}^2}\left[\sum_{j=1}^m \log\mathcal{L}(\beta_i|Y_{i,j},X_{i,j}^1,\dots,X_{i,j}^d)\right]} = \frac{\kappa_{i,l}}{n_{i,l}}.\end{aligned}$$

As a result, we have shown that

$$\lim_{m\to+\infty} m\cdot n_{i,l}\cdot \mathrm{Var}(\hat{\beta}_{i,l}) = \kappa_{i,l},$$

where $n_{i,l}$ denotes the number of items that share the same coefficient with item i for feature l. □

B.4. Proof of Proposition 3

First, for the aggregated model, we have

$$||X\beta - X\hat{\beta}||_2^2 = ||X\beta - X\beta - X(X'X)^{-1}X'\epsilon||_2^2 = ||P\epsilon||_2^2,$$

where $P = X(X'X)^{-1}X'$ is an idempotent matrix (i.e., a matrix which, when multiplied by itself, yields itself). Since $\epsilon\sim N(0,\sigma^2 I)$, we can write,

$$\frac{||X\beta - X\hat{\beta}||_2^2}{\sigma^2} = \left(\frac{\epsilon'}{\sigma}\right)P\left(\frac{\epsilon}{\sigma}\right),$$

where ϵ/σ is a standard normal vector. Based on Section B11.4 in Greene (2003), the above quantity follows a χ^2 distribution with degrees of freedom equal to rank(P). By the commutativity property of the trace operator, we have

$$trace(P) = tr(X(X'X)^{-1}X') = tr(X'X(X'X)^{-1}) = d_x,$$

which is the column rank of matrix X. If X has full rank, then d_x represents the number of features under the aggregated model.

Based on Lemma 1 in Laurent and Massart (2000), we can use the tail bound of χ^2 distribution on our mean squared errors. For any $\gamma > 0$, we have the following high probability upper bound:

$$\begin{aligned}&\mathbb{P}\left(\frac{||X\beta - X\hat{\beta}||_2^2}{\sigma^2} - d_x \ge 2\sqrt{\gamma d_x} + 2\gamma\right) \le \exp(-\gamma),\\ \Longrightarrow\ &\mathbb{P}\left(\frac{||X\beta - X\hat{\beta}||_2^2}{n\times m} \le \frac{\sigma^2\left(2\sqrt{\gamma d_x} + 2\gamma + d_x\right)}{n\times m}\right) \ge 1 - \exp(-\gamma). \qquad (10)\end{aligned}$$

On the other hand, if we use a simple OLS for each item, we obtain n terms of squared errors $||X_i b_i - X_i\hat{b}_i||_2^2$. Each term is a χ^2 distributed variable with degrees of freedom equal to d. Thus, when computing the MSE for the decentralized model, we obtain:

$$\mathbb{P}\left(\frac{\sum_{i=1}^n ||X_i b_i - X_i\hat{b}_i||_2^2}{n\times m} \le \frac{\sigma^2\left(2\sqrt{\gamma\,(nd)} + 2\gamma + nd\right)}{n\times m}\right) \ge 1 - \exp(-\gamma). \qquad (11)$$

Note that $d_x \in [d, nd]$. Therefore, unless all the features are at the SKU level, we have $d_x < nd$, and thus the bound in (10) is tighter than the bound in (11). Consequently, we can achieve a smaller MSE for the aggregated model relative to the decentralized model under the same level of confidence, $\exp(-\gamma)$.

In addition, we can provide a high probability lower bound for the decentralized model. As shown in Laurent and Massart (2000), we have

$$\mathbb{P}\left(\frac{\sum_{i=1}^{n}||X_i b_i - X_i \hat{b}_i||_2^2}{n\times m} \geq \frac{\sigma^2\left(nd - 2\sqrt{\gamma\,(nd)}\right)}{n\times m}\right) \geq 1-\exp(-\gamma). \tag{12}$$

If we compare the upper bound in (10) to the lower bound in (12), we can solve for the sufficient condition under which the aggregated model outperforms the decentralized model with high probability:

$$\begin{aligned} &\frac{\sigma^2\left(2\sqrt{\gamma d_x}+2\gamma+d_x\right)}{n\times m} \leq \frac{\sigma^2\left(nd-2\sqrt{\gamma\,(nd)}\right)}{n\times m}, \\ \Longrightarrow\ & d_x^2-(2A+4\gamma)d_x+A^2 \geq 0, \end{aligned}$$

where $A = nd - 2\sqrt{\gamma nd} - 2\gamma$. As a result, the above inequality holds when

$$d_x \leq nd - 2\left(\sqrt{\gamma nd} + \sqrt{\gamma nd - 2\gamma\sqrt{nd} - \gamma^2}\right)$$

and this concludes the proof of Proposition 3. □